



COURS D'ALGORITHME

SÉRIE 02

NOTIONS SUR LES ALGORITHMES

OBJECTIF PÉDAGOGIQUE :

À la fin de cette série, le stagiaire doit être capable d'appliquer les notions de construction d'un algorithme

PLAN DE LA LEÇON :

I- ENCHAINEMENTS D'ACTIONS

II- ALGORITHME ET PROGRAMME

III- VARIABLES ET AFFECTATIONS

**IV- DONNÉES ET RÉSULTATS : INSTRUCTIONS DE
LECTURE ET ÉCRITURE**

EXERCICES D'APPLICATION

CORRECTION DES EXERCICES

I- ENCHAINEMENTS D'ACTIONS :

Une recette de cuisine bien que n'étant pas un algorithme, permet de préciser l'idée d'algorithme : suite d'actions ou d'instructions pour faire quelque chose (gâteau) avec certaines données (farine, œufs, beurre), et certains outils (Moule, Four, Cuillère).



Il s'agit dans cette phase d'indiquer dans quel ordre doivent être exécutées les actions, et de les exprimer sous un formalisme que l'on peut situer entre le langage naturel et les langages de programmation. Ce formalisme représente en quelque sorte un outil qui permettra une meilleure compréhension entre le concepteur du programme et le programmeur.

Une action est un événement prenant place pendant une période de temps fini et produisant un résultat connu et bien défini. Ainsi, on peut traduire un Algorithme comme étant une liste ordonnée d'opérations ou d'actions, agissant sur un ensemble d'objet et qui permettent de résoudre un problème.

Aussi, l'algorithme étant un procédé de résolution de problème, il peut s'exprimer sous forme d'une suite d'actions plus ou moins élémentaires en fonction des possibilités des personnes ou de la machine qui exécutera l'algorithme.

Exemple :

L'algorithme d'EUCLIDE, qui calcule le PGCD Z de deux (02) entiers a et b , peut être écrit de la façon suivante :

- Effectuer la division de a par b : $a = b q + r$, $0 \leq r < b$
- Si $r = 0$, alors $Z = b$, sinon remplacer a par b , b par r ,
- Recommencer jusqu'à trouver un reste $r = 0$.

Dans cet algorithme les actions sont la division et le remplacement.

II- ALGORITHME ET PROGRAMME :

Nous considérons un ordinateur comme une réalisation concrète d'une machine idéale.

Alors que le programme qui sera exécuté, n'est que l'algorithme écrit pour une machine concrète dans un langage que comprend celle-ci, et qu'on appelle langage de programmation.

D'une autre façon, nous pouvons ajouter que le programme est l'expression destinée à être exécutée par une machine.

Il est donc, nécessaire d'exprimer l'algorithme dans un langage de programmation (tel que FORTRAN, PASCAL, LANGUAGE C,...etc.), il suffit de traduire les expressions algorithmiques (déclaration, action) dans le langage choisi : PASCAL, Langage C, FORTRAN, etc.

Nous avons vu que pour résoudre un problème posé, il faut le décomposer en actions élémentaires :

- **Contexte de définition** : On appelle contexte de définition d'une action, l'ensemble des objets qui apparaissent dans sa description, l'univers de l'algorithme et constitué de l'union des contextes de définition de l'ensemble des actions de l'algorithme.
- **Ensemble fonctionnel** : On appelle ensemble fonctionnel d'un objet, l'ensemble des actions qui le traitent.

Ces deux concepts sont représentés dans l'algorithme respectivement par :

- La partie « déclarations » ;
- La partie « actions ».

Et sont agencés de la manière suivante :

Début

< Partie déclarations >

< Partie actions >

Fin

Les mots **début** et Fin indiquent les limites d'un ensemble qui est dans ce cas l'algorithme lui-même, et qui peut être une partie de l'algorithme qui ressource un sous ensemble particulier.

- **Notion de déclarations** : Nous avons vu que pour être complément un objet doit avoir :
 - **Un nom** qui est l'identifiant de l'objet ou identificateur
 - **Un type** décrivant l'objet
 - **Une valeur** qui est une information liée à l'objet à un instant donné.

Les deux premiers éléments **nom** et **type** sont associés à la déclaration, le troisième est précisé par l'action.

Dans la partie « déclarations », on peut déclarer une constante, comme par exemple : $\pi = 3,14$, ou une variable. La déclaration d'une variable dans un algorithme consiste à associer à cette variable un identificateur et un type.

La forme de la déclaration d'une variable est :

< Identificateur > : < type > ;

Un identificateur permet de référencer l'objet.

Il doit être une chaîne alphanumérique dont le premier caractère est une lettre alphabétique.

Exemple :

MAT, PRENON, MG1, X1 sont des noms d'identificateurs.
1X, 1B2, A, B ne peuvent pas être des noms d'identificateurs.

La déclaration d'objets représente quelques points positifs :
En établissant la déclaration d'objets, nous permet de comprendre un algorithme, de connaître le domaine des valeurs des variables, et par conséquent, elle nous facilite la détection des erreurs de programmation.

III- VARIABLES ET AFFECTATIONS :

L'affectation est une action qui nous permet d'affecter une valeur à une variable et sa forme est :

< IDENTIFICATEUR > := < EXPRESSION

Les symboles ' := ' représente le signe d'affectation et signifie que la valeur de la partie < expression > est affectée à la variable symbolisée par la partie <identificateur> située à gauche du symbole d'affectation.

Exemple 01 :

Y := 3 ; X := 5 ;
X1 := Y + X ; Y1 := X1 + 2 ;
Y1 := Y1 * 2 ; X := Y1 + 3 ;

Dans cette série d'affectations, on remarquera qu'une fois affectée de la valeur 3, Y ne change plus ; par contre X qui vaut 5 après l'affectation de la seconde opération prend une autre valeur après l'affectation de la dernière opération, valeur que le lecteur pourra évaluer aisément.

Soit, dans un algorithme, une variable de nom X.

L'instruction : X := 4 ou X ← 4 signifie, ranger la valeur 4 à la place X (en mathématiques, donner à X la valeur 4)

- Le nom de la variable : Correspond à la place qui lui est réservée :
- Par contre, la valeur correspond au contenu.

Exemple 02 :

Une variable X peut être désignée par une case X ☐

Et quand elle prend la valeur 4 par : X

Exemple 03 :

INSTRUCTIONS	ACTIONS	EFFETS
X := 4 ou X ← 4	X <input type="text"/> ← 4	X 4 <input type="text"/>
X := 5 ou X ← 5	X <input type="text"/> ← 5	X 5 <input type="text"/> Et la valeur 4 a disparu ou écrasée
Y := X ou Y ← X	Y X <input type="text"/> ← 5	Y X <input type="text" value="5"/> <input type="text" value="5"/> La valeur 5 reste dans la case X

IV- DONNÉES ET RÉSULTATS : INSTRUCTIONS DE LECTURE ET ÉCRITURE

1- La lecture: Cette action permet la lecture d'informations ou données à partir de la console ou de tout périphérique d'entrée. Sa forme est:

Ou

LIRE (< liste de variables >) ;

LECTURE (< liste de variables >) ;

Cette action a pour effet d'affecter aux variables définies dans la liste, les valeurs lues sur un périphérique (ex : le clavier) les variables doivent être séparées entre elles par une virgule.

Exemple : lire (X); lire (z,t);

Cette action a le même effet que l'action d'affectation, mais, les valeurs affectées sont des données lues au moment de l'exécution de l'action.

2- L'écriture: Cette action permet d'afficher sur écran ou tout autre périphérique de sortie, la valeur des (ou de la) variable(s).

Sa forme est :

ECRIRE (< RESULTATS >) ;

Un message peut être inclus dans l'ordre d'écriture, il est alors écrit entre côtes ('').

Exemple:

Écrire (X, Y) ;

Écrire (X) ;

Écrire ('La valeur de delta est :' d);

Ce dernier ordre d'écriture aura pour effet:

- L'impression du message : 'la valeur de delta est :'
- L'impression de la valeur de la variable d.

Le message ne subit aucun traitement et il est réécrit tel quel.

Soit l'instruction suivante dans une machine:

$$X := A + 5 \quad \text{ou} \quad X \longleftarrow A + 5$$

Soit une machine qui sait effectuer l'addition et l'affectation.

Cette instruction ne donnera le résultat attendu que si préalablement l'algorithme contient l'instruction lecture de la valeur de A :

Lire (A) ou READ(A) en PASCAL

Il faut donc fournir auparavant à la machine la valeur de A en plus de l'algorithme.

Elle fournira le résultat que si elle est présente dans l'algorithme, l'instruction Ecrire le résultat :

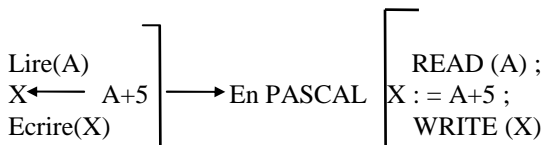
Ecrire(X) ou WRITE(X) en PASCAL.

Ainsi pour :

- **Lire** il faut mettre l'instruction **lecture** ou **READ.**
- **Ecrire** il faut mettre l'instruction **écriture** ou **WRITE.**

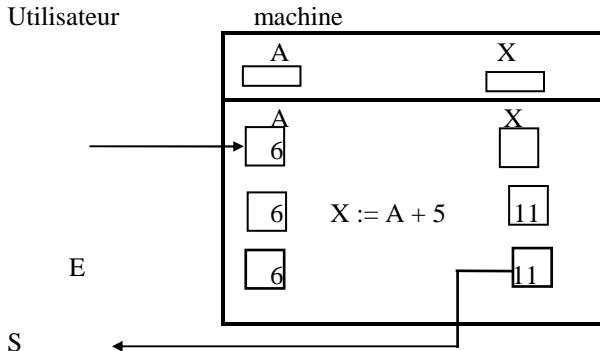
En définitive, l'algorithme sera de la façon suivante:

Ecriture ou **WRITE**



L'exécution de cet algorithme n'aura lieu que si l'utilisateur fournira la valeur de **A**, par exemple 6.

Schéma des actions effectuées par la machine et l'utilisateur



Notre machine est abstraite, l'ordinateur est un modèle concret. Les organes **E** et **S** sont des organes d'E/S de communication entre la machine et l'utilisateur.

1- Résumé :

Nous sommes toujours dans les premières notions de construction d'un algorithme.

À ce stade, les notions fondamentales sont nécessaires pour débiter dans le monde de l'algorithmique.

Ces notions se résument par :

- L'enchaînement des actions dans un algorithme ;
- La différence entre un algorithme et un programme ;
- Les variables et l'action affectation dans un algorithme ;
- Les instructions de lecture et d'écriture pour l'introduction des données et l'affichage des résultats.

EXERCICES D'APPLICATION :

EXERCICE 01:

Compléter le vide :

Un algorithme sert à un procédé

EXERCICE 02:

En suivant la démarche de l'exemple du programmeur d'un lave-linge, essayer de donner l'enchaînement des actions d'un conducteur: de l'automobile.

Quelles sont les opérations à réaliser pour prendre la route ?

EXERCICE 03:

Donner les types de chaque valeur:

15,02 101 1/2 'A' '33' $\sqrt{45}$
21 $\frac{33,51}{\sqrt{3/4}}$ 'AX21' 'bx3/4'

EXERCICE 04:

Écrire un algorithme qui affecte des valeurs réelles aléatoires aux variables A et B puis effectue leur somme. Le résultat de la somme sera quant à lui affecté dans une variable S et affiché à l'écran.

EXERCICE 05:

Écrire le même algorithme mais cette fois-ci les valeurs de A et B seront lues à partir du clavier.

EXERCICE N° 06:

Déterminer les valeurs des variables x, y et z après avoir exécuté l'algorithme ci-dessous:

ALGORITHME EXEMPLE 01;

```
DEBUT
X, Y : ENTIER ;
Z : REEL;
X := 15 ; Y := 10;
    Z := X - Y ; X := X - 5 ;
Y := 1 - Y ; Z := 100 / (19 + Y) ;
    Y := (X - 1) + 1;
FIN.
```

EXERCICE 07 :

Cocher la bonne réponse dans la case correspondante :

1- Un algorithme est-il :

- Un programme exécutable ☐
- La description d'un procédé ☐
- Un programme écrit en un langage de programmation ☐

2- Écrire un algorithme c'est :

- Déterminer la démarche logique d'un problème ☐
- Traduire la démarche en programme ☐
- D'exécuter un programme sur machine ☐

3- L'objectif d'un programme est :

- D'énoncer des valeurs ☐
- De calculer des valeurs ☐
- De décrire des valeurs ☐

4- Dans un algorithme une constante possède :

- Un nom variable, un type fixe et une valeur fixe ☐
- Un nom fixe, un type variable et une valeur fixe ☐
- Un nom fixe, un type fixe et une valeur fixe ☐

5- Dans un algorithme, une variable possède :

- Un nom fixe, un type fixe et une valeur fixe ☐
- Un nom fixe, un type fixe et une valeur variable ☐
- Un nom variable, un type variable et une valeur variable ☐

6- On déclare les constantes ou les variables à l'aide des :

- Instructions ☐
- Valeurs ☐
- Mots clé ☐

EXERCICE 08 :

Répondre aux questions suivantes

- 1- Quelle est la différence entre un algorithme et un programme
- 2- Citer les différents types étudiés, déterminer pour chaque type l'objet manipulé avec des exemples.
- 3- Avec quelle instruction on peut lire une valeur ? (donner trois exemples)
- 4- Quelle est l'instruction qui nous permet d'écrire un résultat ? (donner trois exemples)
- 5- L'instruction d'affectation a pour symbole.....
- 6- Donner le type de chaque valeur :

1001,451.25x $\sqrt{1/\sqrt{1/4}}$, 'A', $\frac{1}{\sqrt{4-1/5}}$, 1, 0, 1/8, $\frac{1}{10^3}$, $5.1e^{-10}$

EXERCICE09 :

Écrire un algorithme permettant aussi de déterminer le volume d'un contenu de longueur LR, de largeur LA et de hauteur HA.

1^{er} Cas : Les valeurs de LR, LA et HA seront lues à partir du clavier

2^{ème} Cas : Les valeurs de LR, LA et HA seront aléatoirement affectées à leurs valeurs respectives.

Enfin, le résultat du volume sera affiché à l'écran.

CORRIGÉ DES EXERCICES :

EXERCICE 01:

Un algorithme sert à décrire un procédé.

EXERCICE 02:

- 1- Prendre les clés
- 2- Introduire la clé dans la serrure
- 3- Déverrouiller la serrure
- 4- Ouvrir la porte
- 5- S'installer dans la voiture.
- 6- Mettre le contact
- 7- Mettre le moteur en marche
- 8- Passer sa vitesse.
- 9- Mettre son clignotant
- 10- Démarrer

EXERCICE 03:

15,02 : Réel 21 : Entier

101 : Entier $\frac{33,51}{\sqrt{3/4}}$: Réel

1/2 : Réel 'A' : Caractère

'AX21' : Caractère '33' : Caractère

'bx3/4' : Caractère $\sqrt{45}$: Réel

EXERCICE 04:

ALGORITHMESOMME1;

DEBUT

A, B: REEL;

A: = 15;

B: = 56;

S: = A + B;

Ecrire (' LA SOMME = ', S);

FIN.

EXERCICE 05 :

ALGORITHME SOMME1;

DEBUT

A, B : REEL;

Lire (A) ;

Lire (B) ;

S := A + B;

Ecrire (' LA SOMME = ', S);

FIN.

EXERCICE 06:

En exécutant les quatre lignes d'instructions on aura :

1/ $X = 15$; $Y = 10$

2/ $Z = 15 - 10 = 5$; $X = 15 - 5 = 10$

3/ $Y = 1 - 9$; $Z = 100 / (19 + (-9)) = 100 / 10 = 10$

4 / $Y = (10 - 9) + 1 = 10$

EXERCICE 07 :

Je coche la bonne réponse dans la case correspondante :

1- Un algorithme est-il :

- Un programme exécutable. ☐
- La description d'un procédé. ☒
- Un programme écrit en un langage de programmation ☐

2- Écrire un algorithme c'est :

- Déterminer la démarche logique d'un problème ☒
- Traduire la démarche en programme ☐
- D'exécuter un programme sur machine. ☐

3- L'objectif d'un programme permet de :

- D'énoncer des valeurs ☐
- Calculer des valeurs ☒
- Décrire des valeurs ☐

4- Dans un algorithme, une constante possède :

- Un nom variable, un type fixe et une valeur fixe ☐
- Un nom fixe, un type variable et une valeur fixe. ☐
- Un nom fixe, un type fixe et une valeur fixe. ☒

5- Dans un algorithme, une variable possède :

- Un nom fixe, un type fixe et une valeur fixe ☐
- Un nom fixe, un type fixe et une valeur variable ☒
- Un nom variable, un type variable et une valeur variable. ☐

6- On déclare les constantes ou les variables à l'aide de :

- Instructions ☐
- Des valeurs ☐
- Des mots clé ☒

EXERCICE 08:

Un algorithme est une méthode d'exploitation de problème permettant d'aboutir directement à la solution, c'est aussi le résultat d'une démarche logique de résolution d'un problème.

Par contre, un programme est un algorithme écrit pour une machine concrète dans un langage de programmation.

Les principaux types étudiés jusqu'à maintenant sont :

- Caractère
- Entier
- Réel
- Logique

Pour le type CARACTERE, les objets manipulés sont tous des caractères alphabétiques, décimaux, signes de ponctuation, le caractère blanc et les caractères spéciaux. Le type ENTIER désigne tous les nombres finis.

Concernant le type REEL, les objets manipulés sont tous les nombres.

Enfin, on peut dire le type LOGIQUE connaît uniquement deux valeurs à savoir : VRAI ou FAUX.

Le mot utilisé pour déclarer une constante est CONSTANCE.

Par contre le mot qu'on utilise pour déclarer les variables est VARIABLE.

On peut écrire une valeur par l'intermédiaire de l'instruction LIRE ().

Exemple : LIRE (A), lire la variable A.

L'instruction qui permet d'afficher ou d'écrire un résultat est l'instruction ECRIRE (). Exemple ECRIRE (X) : permet d'afficher la valeur de X.

L'instruction d'affectation a pour symbole \leftarrow ou :=

Valeurs	Types
1000	Entier
451	Entier
$26x \sqrt{1/\sqrt{1/4}}$	Réel
'A'	Caractère
$1/\sqrt{4}-1/5$	Réel
1	Entier
0	Entier
1/8	Réel
10^3	Réel
5.1E -10	Réel

EXERCICE 09:

Algorithme VOLUME 1;

Début

LR, LA, HA, V : réel;

Écrire ('Donner les valeurs de LR, LA, HA ');

Lire (LR, LA, HA);

$V := LR \times LA \times HA$;

Écrire ('LE VOLUME =', V);

Fin.

Algorithme VOLUME 2;

Début

LR, LA, HA, V : réel;

LR := 4;

LA := 5;

HA := 2;

$V := LR \times LA \times HA$;

Écrire ('LE VOLUME =', V);

Fin.

RÉFÉRENCES BIBLIOGRAPHIQUES :

- 1- **[LAU.85]** Initiation à l'analyse et à la programmation
J.P. LAURENT EDITIONS DUNOD 1985
- 2- **[MEB.84]** Méthodes de programmation B.MEYER ET
C.BAUDOIN Editions EYROLLS 1984
- 3- **[UST.88]** Cours de l'algorithmique. Polycopié de
l'université des sciences et de la technologie
Houari Boumediene USTHB 1988